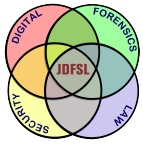

THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 9 | Number 1

Article 3

2014

On Cyber Attacks and Signature Based Intrusion Detection for Modbus Based Industrial Control Systems

Wei Gao
Mississippi State University

Thomas H. Morris
Mississippi State University

ON CYBER ATTACKS AND SIGNATURE BASED INTRUSION DETECTION FOR MODBUS BASED INDUSTRIAL CONTROL SYSTEMS

Wei Gao
Thomas H. Morris
morris@ece.msstate.edu
Department of Electrical and Computer Engineering
Mississippi State University

ABSTRACT

Industrial control system communication networks are vulnerable to reconnaissance, response injection, command injection, and denial of service attacks. Such attacks can lead to an inability to monitor and control industrial control systems and can ultimately lead to system failure. This can result in financial loss for control system operators and economic and safety issues for the citizens who use these services. This paper describes a set of 28 cyber attacks against industrial control systems which use the MODBUS application layer network protocol. The paper also describes a set of standalone and state based intrusion detection system rules which can be used to detect cyber attacks and to store evidence of attacks for post incident analysis. All attacks described in this paper were validated in a laboratory environment. The detection rate of the intrusion detection system rules presented by attack class is also presented.

Keywords: cyber security, intrusion detection system, industrial control system, cyber physical system

1. INTRODUCTION

Industrial control systems, also called Supervisory Control and Data Acquisition (SCADA) control systems, and process control systems, have a strategic importance due to the fact that they are adopted by the critical infrastructure of industrialized nations. There have been several real-world documented incidents and cyber attacks affecting industrial control systems, which clearly illustrate critical infrastructure vulnerabilities. Team Cymru, a specialized Internet security research firm, released a briefing paper (Santorelli, 2009) which discussed malicious port scan activity against their DarkNet (a honey pot) searching for open ports on port numbers commonly associated with SCADA system network protocols. This report showed heavy scanning activity from four areas: Asia, North America, Western Europe and Eastern Europe. The report cited heavy scanning of DNP3 ports from Russia and Taiwan and heavy scanning activities for MODBUS related ports in Western Europe and China. This port scanning is potentially indicative of attackers searching for SCADA systems for later attacks. Stuxnet (Falliere, O'Murchu, & Chien, 2011) is the first known worm to target an industrial control system. Stuxnet targeted computers running the Siemens WinCC SCADA software product. Infected systems had a dynamic link library (DLL) used by the WinCC Step7 tool replaced with a malicious DLL. The worm then monitored communications between the WinCC tool and a remote terminal. If a specific signature related to the remote terminal was found, firmware on the remote terminal was replaced with malicious code. In a third incident in January 2000, a disgruntled engineer attacked the Maroochy Shire Council's sewage control system in Queensland, Australia. A pump in the control system failed to start or stop when specified and alarms failed to alert. This attack caused approximately 264,000 gallons of raw sewage to leak to into nearby rivers (Slay & Miller, 2007). Finally, in 2003, the Davis-Besse nuclear plant in Oak Harbor Ohio was attacked by the Slammer Worm which caused a safety monitoring system of the plant go offline for approximately five hours (Poulsen, 2009).

Forensic systems to detect and store evidence of a cyber attacks against SCADA control systems are not common. Cyber attacks against SCADA control systems may occur at nodes typically found in

enterprise systems such as personal computers, network switches, servers, or firewalls. Cyber attacks may also be directed at devices specific to SCADA control systems such as programmable logic controllers (PLC), programmable automation controllers (PAC), remote terminal units (RTU), master terminal units (MTU), and intelligent electronic devices (IED). This paper addresses a forensic solution to detect attacks against MODBUS clients and servers.

This paper has 2 primary contributions. First, MODBUS is a network communication protocol commonly used in industrial control systems throughout many industries. This paper presents a set of 28 attacks against MODBUS control systems. Attacks are grouped into 4 categories; reconnaissance, response injection, command injection, and denial of service. Each attack is described in detail. All of the attacks presented in this paper were implemented and validated in a laboratory environment against 2 control systems built with commercial hardware and software; a gas pipeline system and a storage tank. Second, this paper presents a state based signature intrusion detection system designed to detect and alert for each of the 28 cyber attacks. All rules were tested in a laboratory setting and this paper provides detection accuracy results. The intrusion detection system and rules described in this paper can be used to detect attacks real time.

The rest of this paper is organized as follows. First, a section on related works is provided. Next, the 28 cyber attacks are presented. Next, the signature based intrusion detection system is discussed. Finally, conclusions and future work are offered.

2. BACKGROUND AND RELATED WORKS

Many attacks against industrial control systems have been described in literature. These attacks highlight the threat to industrial control systems and emphasize the need for tools to capture network traffic related to cyber security attacks. A stealthy attack was developed which steals water from an irrigation canal which used SCADA equipment to track water usage (Amin, Litrico, Sastry, & Bayen, 2013). Multiple cyber attack scenarios including malicious command injection and man-in-the-middle attacks to change process measurements from a wind turbine are presented in (Yan, Liu, & Govindarasu, 2011). The Siemens Simatic S7 PLC has been the subject of extensive review for cyber vulnerabilities. Reconnaissance, fingerprint, replay, authentication bypass, and remote attacks against the Siemens Simatic S7 PLC are presented in (Beresford, 2011). A taxonomy of energy control system vulnerabilities lists probe, flood, bypass, terminate, execute, modify and deletion attacks (Fleury, Khurana, & Welch, 2009). Bulk power transmission systems use state estimation algorithms to plan for power system contingencies. Altered current and voltage measurements in power systems can lead to financial loss and misoperation of the power system (Liu, Reiter, Ning, 2009; Xie, Mo, Sinopoli, 2010).

Digital forensics capabilities for industrial control systems are limited (Nance, Hay, & Bishop, 2009). Most process control systems store significant information about process measurements and process control decisions. Industrial control systems lack forensic tools to capture and store network traffic which may provide evidence of a malicious intrusion. Research is needed to determine the types of information which should be collected and to determine where to place devices to collect information (Valli, 2009). A forensic architecture which identifies locations to collect forensic information for industrial control systems has been proposed (Chandia, Gonzalez, Kilpatrick, Papa, & Shenoi, 2007). The use of the Snort IDS to capture forensic evidence from industrial control system which use the MODBUS network protocol has been proposed in (Slay & Sitnikova, 2009). A data logger to capture and store MODBUS/RTU and MODBUS/ASCII network traffic was proposed in (Morris & Pavurapu, 2010). A retrofit intrusion detection system is described in (Morris, Vaughn, Dandass, 2012). This system was used to demonstrate that 11 of 14 open source Snort rules written for the MODBUS/TCP protocol are also applicable to the MODBUS/RTU and MODBUS/ASCII protocols. This work was extended to include 50 Snort rules to detect protocol mutation attacks against MODBUS servers (Morris, Vaughn, & Dandass, 2013).

The MODBUS protocol (MODBUS-IDA, 2006) is widely used for industrial control systems. This acceptance in industry is partially related to the simplicity of the protocol. MODBUS collectively

refers to MODBUS over Serial Line and MODBUS/TCP protocols.

MODBUS over Serial Line traffic includes two modes; RTU and ASCII. RTU mode packets are binary and transmit a single bit for each bit in an application data unit (ADU). ASCII mode packets convert each byte transmitted to a single ASCII character. Frame delimiters differ for each mode. RTU mode uses dead space on the line to delimit packets, while ASCII mode uses reserved ASCII characters to delimit the start and end of frames. RTU mode appends a cyclic redundancy code (CRC) to the end of each frame. ASCII mode packets end with a linear redundancy code (LRC). The MODBUS protocol data unit (PDU) includes a function code and payload. The function code specifies the type of transaction. Payload contents are specific to the function code.

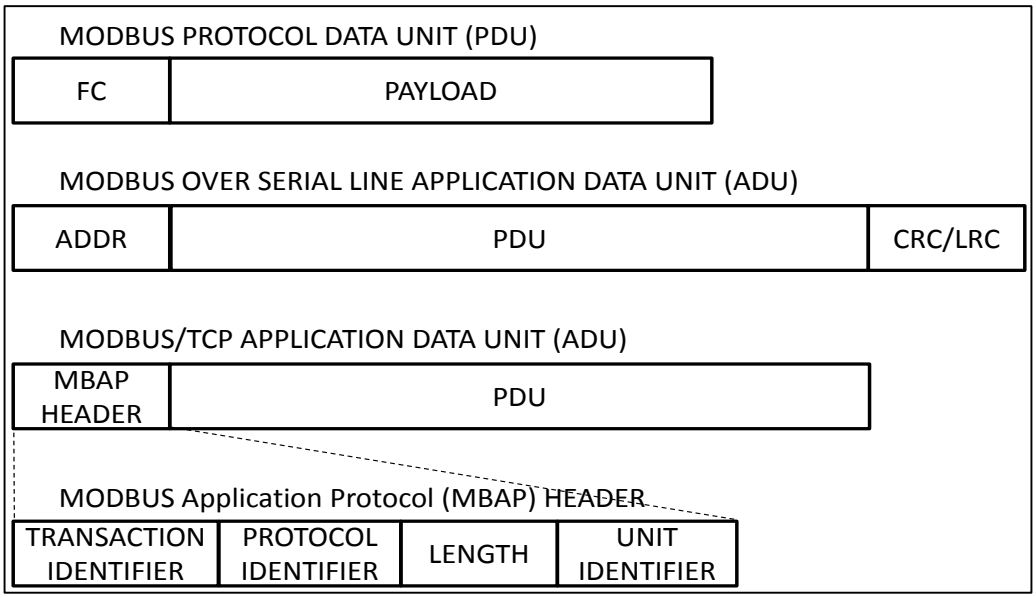

Figure 1 MODBUS Protocol Data Unit (PDU) and Application Data Units (ADU)

MODBUS/TCP is a transmission control protocol (TCP) based protocol intended for use on routable user datagram protocol (UDP) or internet protocol (IP) networks. The PDU for MODBUS/TCP is the same as the PDU for MODBUS over Serial Line. The MODBUS/TCP PDU is prepended with a MODBUS application protocol (MBAP) header which includes a transaction identifier, protocol identifier, length, and unit identifier. The transaction identifier is a unique value for each query and response pair. This is usually a counter. The protocol identifier specifies the protocol version and is always 0. The length parameter specifies the length in bytes of the rest of the packet. The unit identifier is a unique value corresponding to the target slave. The unit identifier is analogous to the MODBUS over serial line address. MODBUS/TCP does not include a CRC. Rather, transmission error detection is provided by the TCP packet. Figure 1 shows the MODBUS PDU, MODBUS ADU for MODBUS over Serial Line and MODBUS TCP, and the MODBUS MBAP header.

The MODBUS protocol is very simple. Generally MODBUS packets come in pairs. A master node sends a query and a slave node sends a response. This query/response paradigm is the most common form of communication. Broadcast packets are allowed in which case there is no response.

Neither MODBUS over Serial Line or MODBUS/TCP include features to prevent replay attacks or to provide a method to authenticate the sender. MODBUS vulnerabilities have been well discussed in literature. The MODBUS/TCP protocol has been the subject of multiple vulnerability studies. MODBUS/TCP systems are vulnerable to denial of service attacks from compromised human machine interface hosts and man-in-the-middle attacks due to a lack of a digital signature to sign network frames (Mallouhi, Al-Nashif, Cox,

Chadaga, & Hariri, 2011). The lack of digital signature or other means to ensure the integrity of network frames in industrial control systems leads to many vulnerabilities. Altered process measurements can lead operators to take incorrect control actions based on malicious fake data and injected commands can cause systems to take unwanted control actions (Huang, et al., 2009; Sridhar, & Manimaran, 2010). A separate comprehensive attack taxonomy with some overlap with this paper is available in (Huitsing, Chandia, Papa, & Shenoi, 2008). In summary, cyber penetration of control systems monitoring and controlling MODBUS based industrial control systems can lead to loss of the visibility and control.Industrial control systems implement feedback control loops to monitor and control the systems. Figure 2 shows a typical industrial control system configuration with three feedback control loops. The first feedback control loop connects a programmable logic controller (PLC) to sensors and actuators which in turn connect to the physical process. This feedback control loop does not use network protocols. The connections are made using analog and digital inputs and outputs on the PLC. The PLC implements a program to perform distributed control actions. For safety reasons the PLC can typically control the physical process without a network connection to the human machine interface (HMI) or master terminal unit (MTU) or other upstream components. The second feedback control loop is from the PLC to the HMI or MTU. The HMI/MTU are typically connected to the PLC with a network which may be implemented via many physical layers (Ethernet, Serial, wireless, etc.) and many transport, network, and application layer protocols (TCP/IP, RS-232, Zigbee, MODBUS, DNP3, etc.). The HMI/MTU continually queries the PLC for sensor measurements. The HMI/MTU may implement a system level control algorithm. The last feedback control loop is the presentation of process information to a human operator. The human in presented with system state information and the human provides supervisory control such as process limits, system state, system control scheme.

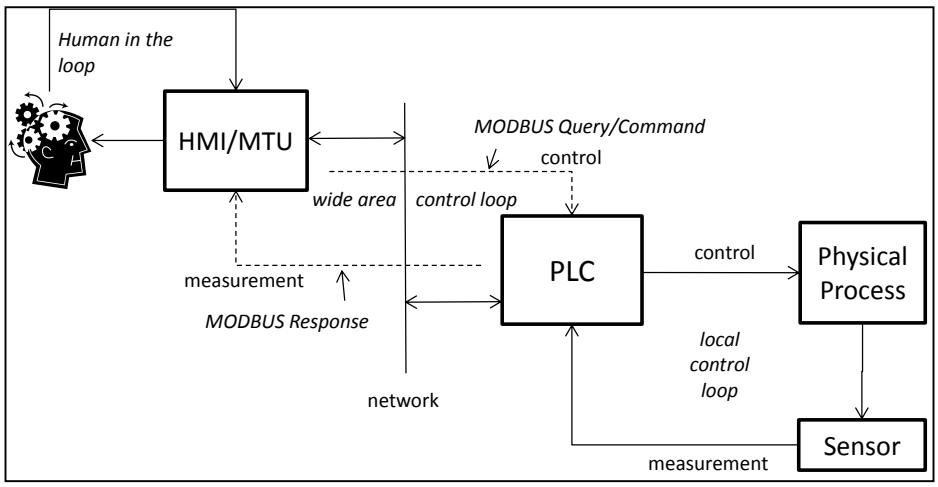

Figure 2 Typical Industrial Control System Feedback Control Loops

3. ATTACKS AGAINST INDUSTRIAL CONTROL SYSTEMS

The set of attacks described in this paper are grouped into four categories; reconnaissance, response injection, command injection, and denial of service. Reconnaissance attacks gather control system network information, map the network architecture, and identify the device characteristics such as manufacturer, model number, supported network protocols, system address, and system memory map. Response injection attacks attempt to present invalid sensor information, process measurements and process state to the controller of a feedback control loop. Command injection attacks attempt to inject invalid commands which cause incorrect control actions. Denial of service attacks attempt to disrupt or break the communication links which implement the feedback control loops. This paper is limited to attacks to the link between the HMI/MTU and the PLC. For the work in this paper this link was implemented with the MODBUS application layer protocol.

Signature based Snort intrusion detection rules were written to detect the attacks described in this section.

Table **1** and Table 2 provide a cross reference for attacks versus associated IDS rule for each attack. The remainder of this section describes the attacks listed in Table 1 and Table 2.

Table 1 Reconnaissance and Response Injection Attacks versus IDS Rule Cross Reference

Name	Associated Intrusion Detection Rule
Address Scan	3.1 RTU/ASCII INVALID ADDRESS 3.2 TCP INVALID ADDRESS
Function Code Scan	3.3 FUNCTION CODE SCAN
Device Identification	3.4 DEVICE ID SCAN
Points Scan	3.5 WRITE POINTS SCAN 3.6 READ POINTS SCAN 3.7 INVALID ADDRESS
Memory Dump	3.6 READ POINTS SCAN
Naïve Read Payload Injection	3.16 INVALID RESPONSE PAYLOAD CONTENT
Invalid Read Payload Size	3.16 INVALID RESPONSE PAYLOAD CONTENT
Naïve False Error Response	3.18 INVALID ERROR RESPONSE
Negative Sensor Measurement(s)	4.1 Pipeline Negative Pressure Response 4.2 STORAGE TANK FILL LEVEL NEGATIVE RESPONSE
Sensor Measurement Grossly Out of Bounds	4.3 Pipeline Measurement Out-of-bounds 4.4 STORAGE TANK MEASUREMENT OUT-OF-BOUNDS
Sporadic Sensor Measurement Injection	4.5 PIPELINE MEASUREMENT MAX RATE OF CHANGE 4.6 STORAGE TANK MEASUREMENT MAX RATE OF CHANGE
Random Sensor Measurement Injection	4.5 PIPELINE MEASUREMENT MAX RATE OF CHANGE 4.6 STORAGE TANK MEASUREMENT MAX RATE OF CHANGE
Constant Sensor Measurement Injection	4.9 PIPELINE CONSTANT MEASUREMENT
High Slope Measurement Injection	4.5 Pipeline Measurement Max Rate of Change 4.6 Storage Tank Measurement Max Rate of Change
Low Slope Measurement Injection	4.7 Pipeline Measurement Min Rate of Change 4.8 Storage Tank Measurement Min Rate of Change
Calculated Measurement Injection	4.5 Pipeline Measurement Max Rate of Change 4.6 Storage Tank Measurement Max Rate of Change 4.7 Pipeline Measurement Min Rate of Change 4.8 Storage Tank Measurement Min Rate of Change
Replayed Measurement Injection	4.5 Pipeline Measurement Max Rate of Change 4.6 Storage Tank Measurement Max Rate of Change 4.7 Pipeline Measurement Min Rate of Change 4.8 Storage Tank Measurement Min Rate of Change

Likely threat actors for the attacks described in this section are either insiders with access to the network attached to the master or slave or malware which penetrates the network and installs malicious code on a machine on the vulnerable network. For MODBUS over Serial Line systems, attacks will generally originate from the computer which hosts the HMI or from an infected PLC. For MODBUS/TCP systems, the primary target would be any computer reach the master and slave devices via network communication. TCP networks often include firewalls which use access control lists to limit traffic on a network. In this case, an attack can only be launched from computers with permission to communicate with the MODBUS master, slave or other device in the feedback loop (such as a historian). Many industrial control systems implement communication links using wireless technologies. These wireless technologies, whether proprietary or standardized, are vulnerable to attack (Reaves & Morris, 2012). Computers which penetrate wireless links can launch attacks against both MODBUS over Serial Line and MODBUS/TCP systems.

Table 2 Command Injection and Denial of Service Attacks versus IDS Rule Cross Reference

Name	Associated Intrusion Detection Rule
Altered System Control Scheme	4.10 Pipeline High Pressure Critical State 4.11 Pipeline Low Pressure Critical State 4.12 Storage Tank High Liquid Level Critical State 4.13 Storage Tank Low Liquid Level Critical State
Altered Actuator State	4.10 Pipeline High Pressure Critical State 4.11 Pipeline Low Pressure Critical State 4.12 Storage Tank High Liquid Level Critical State 4.13 Storage Tank Low Liquid Level Critical State
Continually Altered Actuator State	4.10 Pipeline High Pressure Critical State 4.11 Pipeline Low Pressure Critical State 4.12 Storage Tank High Liquid Level Critical State 4.13 Storage Tank Low Liquid Level Critical State
Altered PID Parameter(s)	3.8 Invalid PID Parameter 4.10 Pipeline High Pressure Critical State 4.11 Pipeline Low Pressure Critical State 4.12 Storage Tank High Liquid Level Critical State 4.13 Storage Tank Low Liquid Level Critical State
Altered Control Set Point	3.9 Pipeline Invalid Set Point 3.10 Storage Tank Invalid Set Point 4.10 Pipeline High Pressure Critical State 4.11 Pipeline Low Pressure Critical State 4.12 Storage Tank High Liquid Level Critical State 4.13 Storage Tank Low Liquid Level Critical State
Force Listen Only Mode	3.11 Force Listen Only Mode
Restart Communication	3.12 Restart Communication
Clear Data Log	3.13 Clear Communications Event Log
Change ASCII Input Delimiter	3.14 Change ASCII Input Delimiter
Invalid Cyclic Redundancy Code (CRC)	4.14 Invalid CRC Count
MODBUS Slave Traffic Jamming	4.15 MODBUS Flood

3.1 Gas Pipeline and Storage Tank Test Bed

The attacks and rules presented in this paper were developed and tested in laboratory environment (Morris, et al., 2011). Testing was performed on two laboratory scale control systems; a water storage tank and a gas pipeline system. The water storage tank models an oil storage tank control system used to monitor oil inventory and distribute oil to refinery processes. The pipeline control system models a pipeline used to move natural gas or other petroleum products to market. Both control systems include a master terminal unit (MTU) and remote terminal unit (RTU) connected with a MODBUS/RTU network. Operators can remotely monitor and control both systems using the HMI. The HMI polls the RTU for system state periodically.

The storage tank control system includes a pump to fill the storage tank, a gravity fed manual relieve valve which allows water to drain from the tank, and a sensor which provides the water level in the primary tank as a percentage of total capacity. The storage tank is an on/off control system which turns the pump on and off to keep the water level between high (H) and low (L) set points.

The pipeline control system contains a closed loop gas pipeline connected to an air pump which pumps air into the pipeline. A solenoid controlled release valve can be opened to release air pressure from the pipeline. A pressure sensor is attached to the pipeline which allows pressure visibility at the pipeline and remotely on an HMI screen. The pipeline uses a proportional integral derivative control scheme to control the pump or solenoid relief valve based upon system configuration.

Ladder logic is used to program the RTU connected to the meters and actuators of both systems. Both systems include registers to store set points which are adjusted via the human machine interface (HMI) software connected to the master node. Pipeline set points include system mode (manual, automatic, or off), manual pump override (on, off), manual relief valve override (open or closed), target pressure, and PID configuration set points. The gas pipeline includes pump state, relief valve state, and pressure measurement output registers. The storage tank RTU ladder logic includes H and L level, HH and LL alarm level, system mode (manual, automatic, or off), and manual pump override (on, off) set points. The storage tank RTU ladder logic also includes 3 output registers which store process parameters; pump state, water level, and alarm state.

To perform the attacks described in this paper a serial bump in the wire was used. This bump in the wire continuously logs all MODBUS over Serial Line traffic. The bump in the wire also includes hooks to alter, delay, or drop packets. The bump in the wire also includes hooks to inject traffic in either direction on the serial link.

3.2 Reconnaissance Class Attacks

Four reconnaissance attacks were implemented for this work. First, the *address scan attack* sends MODBUS queries to all legal MODBUS addresses listening for responses. Implemented MODBUS addresses will respond with an error message or a message indicating success. Non-implemented addresses will lead to no response. The attack implemented for this paper walks through each legal address (0…247) to build a list of implemented addresses. The *function code scan attack* is similar to the address scan except that it walks through all legal MODBUS function code scans to builds a list of implemented function codes by connected address. As with the address scan attack, the function code attack detects implemented function codes based upon device response. Invalid function codes provide a specific invalid function code exception. The *device identification attack* uses two read device identification functions built-in to the MODBUS protocol to learn PLC run status, vendor name, the product code, and the major, minor revision, vendor uniform resource locator (URL), the product name, the model name, the user application name, and other device specific information. This information can be used to search for known vulnerabilities in published vulnerability databases. The *point scan attack* creates a map of implemented MODBUS data block addresses (coils, discrete inputs, holding registers, and input registers) for an identified MODBUS device address. Point scan attacks attempt to read from and write to each legal data block address and use MODBUS response codes to determine which addresses are implemented and which are not. The last reconnaissance attack is the *memory dump attack*. The memory dump attack attempts to read the contents of all PLC data blocks.

3.3 Response Injection Class Attacks

Response injection attacks affect network traffic from MODBUS server to client (responses to MODBUS queries). Response injection attacks take 3 forms. First, response injection attacks can originate from malicious control of a programmable logic controller or remote terminal unit. Second, response injection attacks can capture network packets and alter contents during transmission from server to client. Finally, response injection attacks may be crafted and injected into the network by a third party device.

Naïve malicious response injection (NMRI) attacks lack knowledge of the physical system or its control logic. A *Naïve Read Payload Attack* has a payload made up of random contents, all zeroes, or all ones. This attack is called naïve because no knowledge of the physical system or its control logic is required. The *Invalid Read Payload Size Attack* is an altered or injected MODBUS packet whose payload length does not match the quantity of objects requested by the previous MODBUS query. *The Naïve False Error Response Attack* injects false error responses for valid queries by setting the response function code to the query function code plus 0x80 and providing a valid or invalid MODBUS exception code. The *Negative Sensor Measurement Attack* is an altered or injected MODBUS response packet which includes negative values for measurements which do not typically report negative values. For example, pipeline pressure and water tank percent full should both be positive and a negative value for either measurement would be obviously incorrect. The *Sensor Measurements Grossly Out-Of-Bounds Attack* injects process measurements significantly outside the bounds of alarm set points. Figure 3 is an example of a Sensor Measurements Grossly Out-Of-Bounds Attack. The measurements labeled NMRI are far out of bounds and easy to detect.

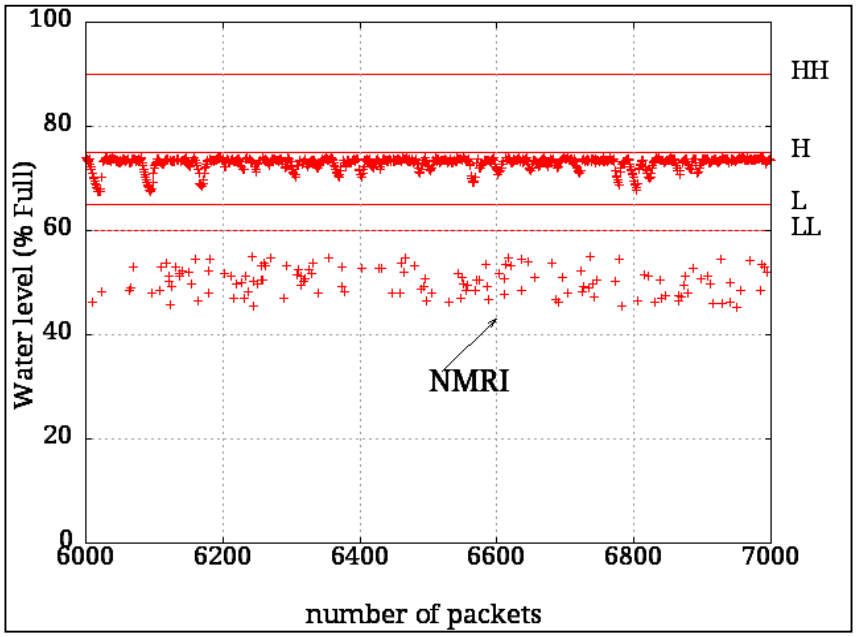

Figure 3 Sensor Measurements Grossly Out-of-Bounds Attack

The *Sporadic Sensor Measurement Injection Attack* injects false process measurements outside the bounds of H and L control set points while not outside the alarm set point range formed by HH and LL. The *Random Sensor Measurement Injection Attack* injects random process measurements for the pipeline pressure and water tank water level. Since these measurement values are random some falsified measurements are within process limits (LL/HH and L/H limits) and some are not.

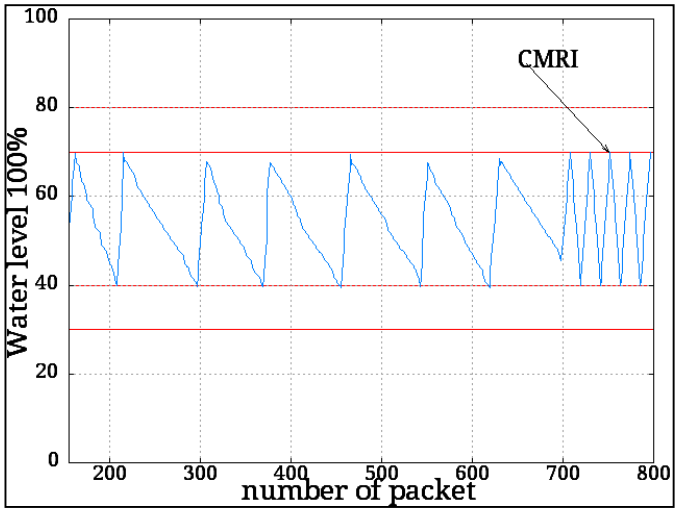

Figure 4 CMRI High Slope Measurement Injection Attack

Complex Malicious Response Injection (CMRI) attacks appear similar to normal system behavior through knowledge of the system being attacked. The *Constant Sensor Measurement* Injection Attack injects malicious packets containing the same measurement over a period of time in order to mask the real state of the system. The Calculated Sensor Measurement Injection Attack injects process measurements using some model of the physical process which is used to predict behavior. The Replayed Measurement Injection Attack injects previously collected measurements captured from eavesdropping on system communications. The replayed measurement and calculated sensor measurement injection attacks are quite similar to one another except for their source of measurement values to inject. The High Frequency Measurement Injection Attack injects measurements in a normal range which changes at a faster rate than the normal system behavior. The Low Frequency Measurement Injection Attack injects measurements in a normal range which changes at a lower rate than the normal system behavior. Figure 4 is an example of a High Frequency Measurement Injection Attack against a storage tank.

3.4 Command Injection Attacks

Command injection attacks inject false control and configuration commands into a control system. Malicious State Command Injection (MSCI) attacks change the state of the process control system abnormally to drive the system from a safe state to a critical state by sending malicious commands to remote field devices. Many control systems include automatic and manual modes. The Altered System Mode Attack injects a command to switch between manual and automatic mode. The Altered Actuator State Attack changes a system actuator state. For example, a pump may be turned on or off or a switch opened or closed. The Continually Altered Actuator State Attack repeatedly alters the state of an actuator in a system. This may be done to attempt to cause physical damage to a system component.

Malicious Parameter Command Injection (MPCI) attacks change parameters used by control schemes on a device to cause incorrect system behavior. The Altered Proportional Integral Derivative (PID) Parameter Attack changes PID control parameters. Changing PID parameters can cause the controller to perform incorrect control actions. The Altered Control Set Point Attack changes device set points. Set points are typically used to provide variable control over a system. For example the storage tank system uses an ON/OFF control scheme to keep the amount of liquid in a tank between a low set point and a high set point.

Malicious Function Code Injection (MFCI) attacks use function codes which specify functions which can be used to cause denial of service attacks or to erase evidence of other attacks. The Force Listen Only Mode Attack injects a command which causes a MODBUS server (a PLC) to no longer transmit on the network. This denial of service will lead to loss of the ability to remotely monitor and control a system. The Restart Communication Attack injects a

command which causes the MODBUS server to restart which leads to a temporary loss of communication. This loss of communication leads to a temporary inability to observe and control the process. The Clear Communications Event Log Attack clears the MODBUS server's communications event log. This attack may be used to erase evidence of a prior attack. The Change ASCII Input Delimiter Attack changes the frame delimiter used for MODBUS ASCII devices to identify the start and end of a network frame. Such a change would cause a denial of service.

3.5 Denial of Service Attacks

Denial of Service (DOS) attacks attempt to break communication links to prevent remote monitoring or control of a system. This section documents two DOS attacks which require large volumes of injected packets. The Force Listen Only Mode, Restart Communication, and Change ASCII Input Delimiter attacks are also denial of service attacks. These three were listed in the command injection attack section because they require an injected MODBUS command to initiate, whereas, the DOS attacks in this section require large volumes of traffic to initiate.

The *Invalid Cyclic Redundancy Code (CRC) attack* injects large volumes of MODBUS packets with incorrect CRC. Packets with invalid CRC are rejected by both MODBUS servers and clients. The victim device must check the CRC of each packet. A flood of packets with invalid CRC can overwhelm a device causing slow system responses or no system responses due to a crashed network stack.

The *MODBUS Slave Traffic Jamming Attack* 18 is specific to a proprietary wireless communication system. The proprietary wireless radio includes a carrier sense back off arbitration scheme which causes legitimate slaves to wait for a clear line to transmit. In laboratory experiments, attackers were able to force a legitimate slave to stay idle ad infinitum by continuously transmitting from a 3rd radio connected to the network. Figure 5 shows the impact of the *MODBUS Slave Traffic Jamming Attack* against a storage tank from the perspective of the HMI. When the attack starts the HMI no longer receives responses to its water level queries and therefore the water level in the plot no longer changes.

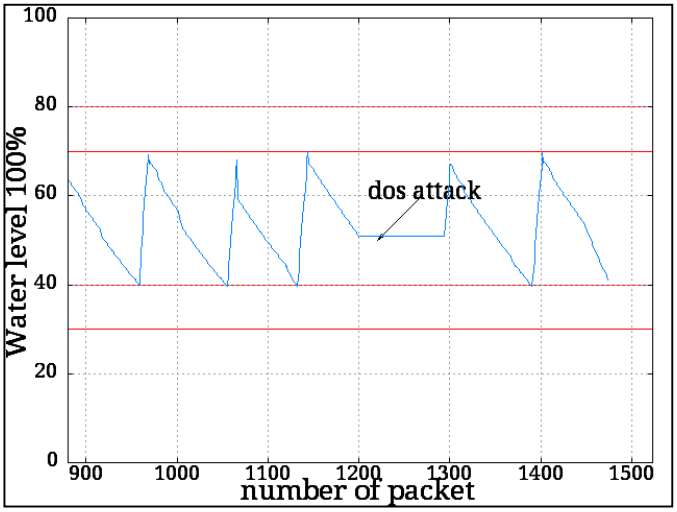

Figure 5 MODBUS Slave Traffic Jamming Attack Effect

4. SIGNATURE BASED INTRUSION DETECTION FOR INDUSTRIAL CONTROL SYSTEMS

This section provides a set of rules for a signature based intrusion detection system. The rules described in the section are designed to detect the attacks described previously. The rules are divided into two types; standalone and state-based rules. The set of rules developed for this work are listed in Table 3 and Table 4. Standalone rules parse a single MODBUS packet looking for a match to a specific

signature. If the signature is present in the parsed packet then the packet is classified as a match and an alert is issued. The standalone rules are implemented using the Snort intrusion detection tool. The second type of rule is called state based. State based rules require knowledge from previous MODBUS packets or from another source, such as a process sensor. This extra knowledge may be related to the protocol state or the state of the industrial control system being monitored. State based rules are processed using a Snort pre-processor, hence forth referred to as the state based layer. Snort passes the MODBUS payload in its entirety to the state based layer. The state based layer is written in the C programming language. A set of C-language structures were developed to store the state of the MODBUS protocol and a historical model of the state of the industrial control system. The model of the state of the industrial control system is system specific and requires expert knowledge to develop. The protocol state structure stores the last received MODBUS packet. The historical state for the industrial control system holds the command state and the process state. The command state is updated each time a command is sent to a MODBUS server. For the pipeline, the command state includes items such as the on/off state of the pump, the open/closed state of the relief valve, the system mode (manual or automatic), copies of set points, and other process specific control information. The process state includes measurements related to the process. For the pipeline, the process state includes the last pressure reading and other system measurements. Figure 6 shows the intrusion detection system architecture with separate standalone and state based layers. For this work, a tap cable was added which allowed monitoring of all MODBUS over Serial line traffic. The tap cable includes only a receive pin and therefore cannot transmit. The tap cable monitored traffic between the MODBUS master PLC and slave PLC (aka RTU). A gasket which converts MODBUS over Serial Line traffic to MODBUS/TCP traffic was used to feed traffic to Snort for monitoring. For MODBUS/TCP systems a port mirror can be used to capture traffic on the network.

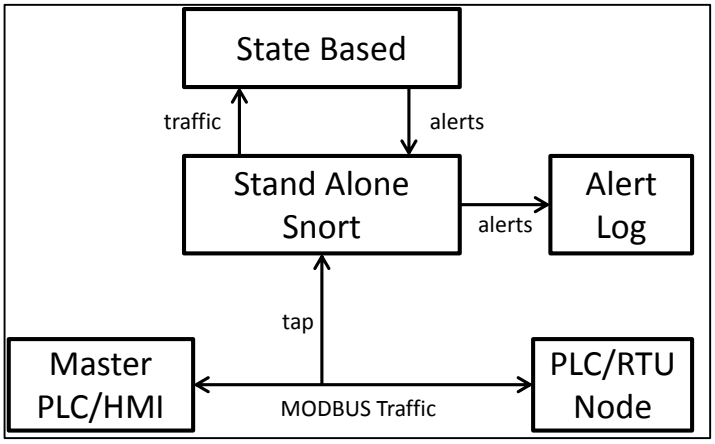

Figure 6 Intrusion Detection System Architecture

4.1 Snort Rule Descriptions

In this section and in Table 1 and Table 2 signature based IDS rules are referred to using the notation *table number dot rule number (t.r)*. For example Rule 3.1 is the first rule in Table 3 and Rule 4.2 is the second rule in Table 4.

Table 3 lists the standalone rules. Rules 3.1-3.7 are used to detect reconnaissance attacks. MODBUS address scans differ based upon the upper network layers. MODBUS/TCP address scans search for IP addresses with MODBUS servers. MODBUS RTU and ASCII use a 1 byte address field. MODBUS RTU and ASCII address scans search for address values which provoke a response. MODBUS systems typically have a static set of member nodes each with a fixed address. A white list of system address, whether IP addresses or 1 byte addresses for MODBUS RTU and ASCII systems, can be developed. Rules 3.1 and 3.2 are used to detect

packets which are addressed to a system not in address white lists.

A system specific white list can be developed which includes all legal function codes a MODBUS server supports. The function code white list can include all public MODBUS function codes and any user defined function codes. However, many systems use only a subset of the public MODBUS function codes. As such the function code white list should be limited to function codes used by the system being protected. Rule 3.3 alerts if a packet is received which includes a function code not in the function code white list.

Device ID scans are detected with Rule 3.4 which alerts if the function code is 0x11 or 0x2B. Since the allowed function code white list will not include function codes 0x11 and 0x2B Rule 3.3 will also alert for Device ID attacks. Care should be taken when implementing the device ID scan rule since it effectively bars operators from reading device IDs. If reading the device ID is required the device ID scan rule should be updated accordingly.

Point scans are detected using 3 rules. First, Rules 3.5 and 3.6 use address white lists to detect read and write transaction addresses for non-existing or black listed memory regions. There is a write-able regions white list and a read-able regions white list. These white lists include a list of contiguous address regions which allow memory writes and reads respectively. Rules 3.5 and 3.6 rules confirm the start address is within the white listed address region. Rules 3.5 and 3.6 also compute the end address and confirm the end address is within the white listed address region. The end address is computed by adding the read or write length to the start address. The length value is available for all write and read function codes. Rules 3.5 and 3.6 are implemented as separate rules for each write and read function code. They are described here as two rules for simplicity. Rule 3.7 is also used to detect point scans. Rule 3.7 alerts if an invalid address exception code is detected for a read or write function code. Properly functioning MODBUS clients should not attempt to access an address which is not configured for reading on the target MODBUS server. As such any instance of the invalid address exception code is evidence of a points scan. Rule 3.7 is not adequate to detect all point scans as it is possible certain memory regions are available on a MODBUS server by default while not actually in use for the specific control system being protected. The white lists associated with Rules 3.5 and 3.6 should only include address regions needed for the specific control system being protected. Rule 3.6 is also used to detect memory dump attacks. The read address white list should only include read addresses reserved for access through the network and should not include addresses used for internal program variables.

Rule 3.8 alerts when a command to set PID parameters to an invalid value is detected. The pipeline system uses a PID controller to open and close the relief valve to keep the pressure within the high (H) and low (L) set points. Small changes in PID parameters can lead to drastic changes in system behavior. As such, Rule 3.8 alerts for any change in any of the five PID parameters; gain, rate, reset, dead band, and cycle time. In systems where PID values are regularly changed, this type of rule can include valid ranges for PID values.

Table 3 MODBUS Standalone Intrusion Detection Rules

Number	Name	Description
1	RTU/ASCII Invalid Address	packet address ∉address white list
2	TCP Invalid Address	IP address ∉IP address white list
3	Function Code Scan	packet function code ∉ function code white list
4	Device ID Scan	function code ∈ {0x11, 0x2B}
5	Write Points Scan	(start address && end address)∈ write address region white list
6	Read Points Scan	(start address && end address)∈ read address region white list
7	Invalid Address	Response & function code ∈{0x81, 82, 83, 84, 85, 86, 8F, 90, 95, 96, A2, A3, A4} & exception code = 02
8	Invalid PID Parameter	gain != 115 | rate != 0 | reset != 0.2 | deadband != 0.5 | cycle time != 1.0
9	Pipeline Invalid Set Point	P > 30 | L < 5
10	Storage Tank Invalid Set Point	H > 80 | L < 20 | HH > 90 | LL < 10
11	Force Listen Only Mode	function code = 0x08 & sub-function = 0x0004
12	Restart Communication	function code = 0x08 & sub-function = 0x0001
13	Clear Communications Event Log	function code = 0x08 & sub-function = 0x0001& data = 0xFF00
14	Change ASCII Input Delimiter	function code = 0x08 & sub-function = 0x0003
15	Illegal Packet Length	Packet length > 255
16	Invalid Response Payload Content	Response & function code is 0x80 & each byte in payload is 0x00 or 0Xff
17	Invalid Response Payload Size	Response & function code is 0x80 & payload size != request payload size in the command
18	Invalid Error Response	Function code is 0x80 & exception code !∈{0x1, 2, 3, 4}

Table 4 MODBUS State Based Intrusion Detection Rules

Number	Name	Description
1	Pipeline Pressure Negative Response	Response & function code = 03& point address = 04 & value <0
2	Storage Tank Fill Level Negative Response	Response & function code = 03 & point address = 07 & value <0
3	Pipeline Measurement Out-of-bounds	Response & function code = 03& point address = b7 & (value < LL OR value > HH)
4	Storage Tank Measurement Out-of-bounds	Response & function code = 03 & point address = e9 & (value < LL OR value > HH)
5	Pipeline Measurement Max Rate of Change	Response & function code = 03 & point address = b7 & slope > SL_{max} & slope != 0
6	Storage Tank Measurement Max Rate of Change	Response & function code = 03& point address = e9 & $(M_1\text{-}M_0/T_1\text{-}T_0) > SL_{max}$ & slope != 0
7	Pipeline Measurement Min Rate of Change	Response & function code = 03 & point address = b7 & slope < SL_{min} & slope != 0
8	Storage Tank Measurement Min Rate of Change	Response & function code = 03& point address = e9& slope < SL_{min} & slope != 0
9	Pipeline Constant Measurement	Response & function code = 03& point address = b7& NC > T & SystemMode = AUTO
10	Pipeline High Pressure Critical State	pressure > HH & pump = ON & relief valve = CLOSED
11	Pipeline Low Pressure Critical State	pressure < LL & (pump = OFF | relief valve = OPEN)
12	Storage Tank High Liquid Level Critical State	liquid level > HH & pump = ON
13	Storage Tank Low Liquid Level Critical State	liquid level < LL & pump = OFF & system = ON
14	Invalid CRC Count	invalid CRC count > 5 in time window 5
15	MODBUS Flood	MODBUS Packet Count > 5 in time window 5

Rules 4.1 and 4.2 alert when sensor measurements are negative. These rules are system specific. Rule 4.1 alerts for a negative pressure measurement for a pipeline system. Rule 4.2 alerts for a negative water level measurement for a storage system. In each case, the rule must be programmed with the function code used to read the measurement, the exact address of the point which stores the measurement the measurement, and the width of the measurement in bytes. There may be more than one function code used to poll the MODBUS server for a measurement. In this case, multiple instances of the rule can be created to cover each case. Also, often MODBUS clients will read many points in a single read. In this case, Rules 4.1 and 4.2 should be updated to check the correct set of bytes within the larger read payload. These rules are state-based because MODBUS read responses do not include the address which was read from. As such, the state based layer stores the read command details, including the read

start address and quantity, and uses this information within the rule.

Rules 4.3 and 4.4 alert when process measurements are grossly out of bounds. These rules are system specific. Rule 4.3 alerts for an out-of-bounds pressure measurement for a pipeline system. Rule 4.4 alerts for an out-of-bounds water level measurement for a storage system. Both rules are programmed with extreme limits for their respective process. These rules may be configured as rules if the extreme limits for the process are static or are changed infrequently. The extreme limits should be set relative to the alarm thresholds for the process measurement.

Rules 4.5-4.8 alert if the rate of change of a sensor measurement exceeds or falls below specific maximum and minimums respectively. Rules 4.5 and 4.6 alert for high rates of change. A high rate of change may be a symptom of a high slope attack or sporadic or random measurement injection attacks. As successive measurements are observed in network traffic, the most recent measurement value and the timestamp of the latest measurement are stored in the state based layer. As new measurements are observed, the rate of change of process measurements is calculated and compared with a predefined maximum rate of change value. The rate of change is calculated using equation 1, where M0 is the current measurement, M1 the previous measurement, T1 is the time stamp of the current measurement, and T0 is the timestamp of the previous measurement. Control systems tend to poll sensor measurements periodically meaning T1-T0 will be approximately constant. For Rules 4.5 and 4.6, the term *SLmax*, in Table 4, is the maximum rate of change allowed. In some systems, separate maximum rates of change will be defined for the rising and falling measurement case. In such cases, two rules will be needed for each system. The maximum rate of change is system specific and must be defined in consultation with a system expert. This rule will not detect all sporadic and random sensor measurement injections since some injected measurements may be close enough to the previous value to not trigger the alert. However, during an extended attack many measurements will trigger an alert.

$$slope = \left| \frac{M_1 - M_0}{T_1 - T_0} \right| \quad (1)$$

Rules 4.7 and 4.8 alert for low rates of change. These rules detect low slope measurement injection attacks. Similar to the maximum rate of change, the minimum rate of change is system specific and system experts should be consulted when setting this limit. In many systems, acceptable minimum rate of change cannot be defined. In such cases, no minimum rate of change rule should be applied. Also many systems have a minimum rate of change with the exception that no change is an allowed condition. For this work, Rules 4.7 and 4.8 do not alert if the calculated slope is 0.

Rule 4.9 alerts when a threshold of *T* consecutive packets is observed with the same process measurement. In the description of Rule 4.9, from Table 4, the variable *NC* is the count of consecutive packets without a measurement change. The state based layer calculates the count of consecutive packets without a process measurement change. The state based layer also stores the current process system mode (AUTO or MANUAL). Rule 4.9 only detects constant level injection attacks in system state in which the process measurement is known to change. For systems where the measurement may legally be constant, these rules are not applicable. A constant level injection attack may also be detected by the rate of change rules (Rules 4.6 and 4.7) if the injected measurement varies significantly from the measurement observed immediately before the attack initiates. Subsequent packets during the attack will not trigger alerts from the rate of change rules.

Rules 4.10-4.13 monitor the physical process state and alert when the process is in a critical state. For this paper, a critical state is defined as a state of alarm in which the control settings will drive the system further away from a normal system state. Rule 4.10 alarms if the pipeline pressure is above the high level alarm set point (HH) and the pump is on and the relief valve is closed. Rule 4.11 alarms if the pipeline pressure is below the low level alarm set point (LL) and the pump is off or the relief valve is open. Rule 4.12 alarms if the storage tank liquid level is above the alarm set point (HH) and the pump is on. Rule 4.13 alarms if the storage tank liquid level is below the alarm set point (LL) and the pump is off

and the system is on. Each of these states should never occur. These IDS rules may alert due to an actual process fault which leads the system to a critical state or may occur due to a cyber-attack driving the process to a critical state via command injection attack. These state based rules will alert for some but not all cases of the altered system control scheme, altered actuator state, continually altered actuator state, altered proportional integral derivative parameter(s), and altered control set point command injection attacks. The rules will only alert if the command injection drives the process to a critical state.

Rule 4.14 detects the invalid cyclic redundancy code (CRC) flood attack. MODBUS-RTU mode uses a 16-bit CRC and MODBUS-ASCII uses an 8-bit longitudinal redundancy code (LRC). The functions to generate the MODBUS-RTU and MODBUS ASCII CRC and LRC respectively are available in the MODBUS over Serial Line Specification and Implementation Guide 9. Each packet is monitored in the state based layer. The CRC/LRC is calculated within the state based layer and compared to the CRC/LRC with the packet. A count of failed CRC/LRC over a time window is kept. If the number of failed CRC/LRC exceeds a programmable threshold an alert is issued. For this work, the time window was 1 seconds and the number of failed CRC required to trigger the rules was 2.

The *MODBUS Slave Traffic Jamming Attack* is detected with Rule 4.15. If the count of MODBUS packets in a given time window exceeds a threshold a flood alert is produced.

Table 5 Signature based IDS Detection Results by Class

	Pipeline System		**Storage System**	
	Detection Rate	**False Positive**	**Detection Rate**	**False Positive**
Reconnaissance	98.7%	0.0%	98.7%	0.0%
NMRI	95.4%	0.8%	94.2%	0.8%
CMRI	92.5%	0.5%	93.7%	0.6%
MSCI	89.8%	0.7%	90.1%	0.7%
MPCI	93.1%	0.0%	93.0%	0.0%
MFCI	100%	0.0%	100%	0.0%
DOS	100%	0.0%	100%	0.0%
Normal	99.5%	-	99.5%	-

4.2 Snort Rule Validation

Table 5 lists signature detection rate and false positive percentage for signature evaluation using the pipeline and storage systems. The results in Table 5 are presented by attack class.

The detection rate for reconnaissance attacks was 98.7% for the pipeline system and 98.7% for the storage tank system. A review of the misclassified attacks showed that the signature based IDS failed to detect the malicious packets that contain valid device ID, function codes, and read/write memory addresses. During a reconnaissance attack, attackers scan ranges of device addresses, function codes and memory addresses. These ranges contain valid device addresses, function codes and memory addresses which have been defined in white lists. The rules miss these malicious packets and do not trigger alarms. The false positive rate for the reconnaissance attacks was 0% for the pipeline control system and 0% for the storage tank control system. This means the signature based IDS does not misclassify the normal traffic or other types of attack as reconnaissance attacks.

The detection rate for NMRI attacks was 95.4% for the pipeline system and 94.2% for the storage tank system. A review of the misclassified attack cases showed that the signature based IDS failed to detect

malicious packets that contain gas pressure measurements or water level measurements out of the bounds defined by the IDS rules. During the NMRI attack, when the malicious packet contains a false measurement that is very close (in time) to valid or true measurement, the malicious packet will not violate the IDS rules. The false positive rate for the NMRI attacks was 0.8% for the pipeline control system and 0.8% for the storage tank control system. A review of the false positive cases showed that the IDS misclassified some normal traffic packets as NMRI packets. Sometimes valid measurements fall outside the bounds set by Rules 4.3 and 4.4. These situations will trigger a false alarm. The same occurrence will trigger an alarm. Both the rules and alarm levels should be updated to better reflect system behavior.

The detection rate for CMRI attacks was 92.5% for the pipeline control system and 93.7% for the storage tank control system. A review of the misclassified attack case showed two issues. CMRI attack detection heavily relies on the min max rate of process measurement change rules. It is hoped that the transition from normal unaltered measurements to altered measurements has a slope which violates state based Rules 4.6-4.9 thresholds. If an attack lasts multiple packets and within the attack measurements do not violate the state based Rules 4.6-4.9 thresholds those packets will be misclassified. The first packet of the attack may be detected but some packets within the attack may not be detected. Second, if the state based Rules 4.6-4.9 thresholds are not violated at all during the attack then the attack can go unnoticed. This is especially possible during the calculated and replay CMRI attacks. The false positive rate for the CMRI attacks was 0.5% for the pipeline system and 0.7% for the storage tank system. A review of false positive cases showed that the IDS misclassified some normal traffic as CMRI packets. The timestamp applied to packets is not added by the RTU PLC. The timestamp is added by a separate data logger process. The timestamp is sometimes incorrect due to the data logger computer becoming busy with other processes. This leads to incorrect rate of change values which trigger the minimum rate of change rules.

The detection rate for MSCI attacks was 89.8% for the pipeline system and 90.1% for the storage tank system. A review of the misclassified attack cases showed that the IDS failed to detect some packets that contain malicious system state commands. The Snort rules defined ranges of the system parameters. When these crafted parameters were in the allowed range, the malicious packets were not detected. The false positive rate for the MSCI attacks was 0.7% for the pipeline system and 0.7% for the storage tank system. The false positive cases occur when alarm levels are set too close to the normal operating range of the system. These levels can be adjusted to minimize false positives.

The detection rate for MPCI attacks was 93.1% for the pipeline control system and 93.0% for the storage tank control system. A review of the misclassified cases showed that the IDS failed to detect some MPCI packets that contained set point values that do not violate Rules 4.9-4.10. During an MPCI attack, when a malicious packet contains set point values within allowed ranges defined by Rules 4.9-4.10 alerts are not triggered. The false positive rate for the MPCI attacks was 0% for the pipeline control system and 0% for the storage tank control system.

The detection rate for MFCI attacks was 100% for the pipeline control system and 100 % for the storage tank control system. The false positive rate for the MFCI attacks was 0% for the pipeline control system and 0% for the storage tank control system. MFCI attacks require use of specific banned function and sub function codes and are therefore easily detected.

The detection rate for DOS attacks was 100% for the pipeline control system and 100% for the storage tank control system. The false positive rate for the DOS attacks was 0% for the pipeline control system and 0% for the storage tank control system. High volumes of packets are easy to detect.

5. CONCLUSIONS

Industrial control systems are vulnerable to multiple types of network based attacks including reconnaissance, response injection, command injection, and denial of service attacks. This paper presents 28 network based attacks which target MODBUS systems. The attacks were implemented and tested against two control systems; a pipeline and storage tank system. The existence of such attacks drives the need for digital forensic systems which can capture and store evidence of attacks for intrusion detection and post incident forensic

analysis. This paper presents 18 standalone and 15 state based IDS rules to detect the cyber attacks presented. Standalone rules monitor the contents of a single network frame to detect an attack. State based rules monitor multiple network packets to build up model of the present state of the communication protocol or the control system itself. Together standalone and state based rules provide a highly effective means to detect cyber attacks against control systems with low false positive rates.

A great deal of industrial control system research centers around anomaly based and specification based IDS research. Signature based IDS are generally known quantities by the IDS research community and therefore may not be considered for use in industrial control systems. However, industrial control systems have regular communication patterns and predictable system state and control schemes. This regularity and predictability make them prime candidates for signature based IDS. Implementing the rules described in this paper, or similar rules for different types of control systems, can lead to a more secure critical infrastructure. Signature based IDS rules can be added as events occur and can be shared within industry and government to provide a level of protection that currently does not exist. Also, signature based IDS rules are easily adjustable and therefore can be customized for the specific system which they protect. Many control systems are designed and maintained by 3rd party system integrators. System integrators should be trained in the use of signature based IDS to provide a first line of protection for control systems.

REFERENCES

1. Amin, S., Litrico, X., Sastry, S., & Bayen, A. M. (2013a). Cyber security of Water SCADA systems -Part I: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology, 21*(5), 1963-1970.
2. Beresford, D. (2011). Exploiting Siemens Simatic S7 PLCs. Black Hat USA Briefings & Training USA + 2011. July 30–August 4, 2011, Las Vegas, NV, USA.
3. Chandia, R., Gonzalez, J., Kilpatrick, T., Papa, M., & Shenoi, S. (2007). Security strategies for SCADA Networks. *Critical Infrastructure Protection, 253*, 117-131.
4. Falliere, N., O'Murchu, L., & Chien, E. (2001). W32. Stuxnet Dossier, Symantec Tech. Rep. 1.4. Retrieved on April 30, 2014 from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.
5. Fleury, T., Khurana, H., & Welch, V. (2009). Towards a taxonomy of attacks against energy control systems, in *Critical Infrastructure Protection II*, eds. M. Papa and S. Shenoi. Springer.
6. Huang, Y. Cardenas, A., Amin, S., Lin, Z., Tsai, H., & Sastry, S. (2009). Understanding the physical and economic consequences of attacks on control systems. *International Journal of Critical Infrastructure Protection*, *2*(3), 73-83.
7. Huitsing. P., Chandia, R., Papa, M., & Shenoi, S., (2008). Attack taxonomies for the Modbus protocols. *International Journal of Critical Infrastructure Protection*, *1*, 37-44.
8. Liu, Y., Reiter, M., & Ning, P. (2009). False data injection attacks against state estimation in electric power grids. *16th ACM Conference on Computer and Communications Security*, November 9-13, 2009, Chicago, IL, USA.
9. Mallouhi, M., Al-Nashif, Y., Cox, D., Chadaga, T., & Hariri, S. (2011). A testbed for analyzing security of SCADA Control Systems (TASSCS). 2011 IEEE PES Innovative Smart Grid Technologies (ISGT), January 17-19, 2011, Anaheim, CA, USA.
10. MODBUS-IDA. (2006). MODBUS Application Protocol Specification V1.1b. Retrieved on April 30, 2014 from
11. www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf.
12. Morris, T. & Pavurapu, K. (2010). A retrofit network transaction data logger and intrusion detection system for transmission and distribution substations. IEEE International Conference on Power and Energy (PECon). December 2-5, 2012, Sutera Harbour, Sabah, Malaysia.
13. Morris, T., Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., & Reddi, R. (2011). A control system testbed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection*, *4*(2), 88-103.
14. Morris, T., Vaughn, R., & Dandass, Y. (2012). A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control

systems. 45th IEEE Hawaii International Conference on System Sciences (HICSS), January 4-7, 2012, Wailea, Maui, HI, USA.
15. Morris, T., Vaughn, R., & Dandass, Y. (2013). Deterministic intrusion detection rules for MODBUS protocols. 46th IEEE Hawaii International Conference on System Sciences (HICSS), January 7-10, 2012, Wailea, Maui, HI, USA.
16. Nance, K., Hay, B., & Bishop, M. (2009). Digital forensics: Defining a research agenda. 42nd Hawaii International Conference on System Sciences, January 5-8, 2009, Waikoloa, Big Island, HI, USA.
17. Poulsen, K. (2009). Slammer worm crashed Ohio nuke plant network. Retrieved on April 30, 2014 from http://www.securityfocus.com/news/6767.
18. Reaves, B., & Morris, T. (2012). Analysis and mitigation of vulnerabilities in short-range wireless communications for industrial control systems. *International Journal of Critical Infrastructure Protection*, *5*(3-4), 154-174.
19. Slay, J., & Miller, M. (2007). Lessons learned from the Maroochy water breach, in *Critical Infrastructure Protection*, eds. E. Goetz and S. Shenoi. Springer.
20. Santorelli, S. (2009). Who is looking for your SCADA infrastructure? Retrieved on April 30, 2014 from http://www.teamcymru.org/ReadingRoom/Whitepapers/2009/scada.pdf.
21. Slay, J., & Sitnikova, E. (2009). SCADA process control systems security forensics. *Forensics in Telecommunications, Information and Multimedia*, *8*, 77-82.
22. Sridhar, S., & Manimaran, G. (2010). Data integrity attacks and their impacts on SCADA control system. 2010 IEEE Power and Energy Society General Meeting. July 25-29, 2010, Minneapolis, MN, USA.
23. Valli, C. (2009). SCADA forensics with Snort IDS. The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'09). July 13-16, 2009, Las Vegas, NV, USA.
24. Xie, L., Mo, Y., & Sinopoli, B. (2010). False data injection attacks in electricity markets. First IEEE International Conference on Smart Grid Communications (SmartGridComm), October 4-6, 2010, Gaithersburg, Maryland, USA.
25. Yan, J., Liu, C., & Govindarasu, M. (2011). Cyber intrusion of Wind Farm SCADA system and its impact analysis. IEEE/PES Power Systems Conference and Exposition (PSCE), March 20-23, 2011, Phoenix, AZ, USA.